IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-5. (Canceled)

6. (**Currently Amended**) An apparatus comprising a machine-readable storage medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

assigning a first register class to at least one symbolic register in at least one instruction;

assigning a second register class to the at least one symbolic register based at least in part on a linear conjunctive forward dataflow analysis that iterates ~~each~~ **a** basic block of instructions **in a compilation unit** only once;

moving existing register class fixups for the assignment of the second register class **to a different location** and removing unnecessary register class fixups, the moving and removing to reduce the register class fixups, wherein register class fixups are instructions inserted into a program in response to one or more register class assignment modifications; and

renaming the at least one symbolic register,

wherein each instruction includes assignment of one of the first register class assigned and the second register class assigned.

7. (Original) The apparatus of claim 6, said assigning the first register class instruction is an initial assignment.

8. (**Currently Amended**) The apparatus of claim 6, said assigning the second register class further including instructions which, when executed by a machine, cause the machine to perform operations including:

marking a register class assignment map that operates to track register class assignments at a block entry of [[a]] **the** compilation unit;

marking [[the]] **a** register class assignment map at a block exit of the compilation unit;

determining [[the]] **a** register class assignment map at an entry of a**n** instruction in a block **of the compilation unit**; and

determining [[the]] **a** register class assignment map at an exit of [[a]] **the** instruction in the block **of the compilation unit**.

9. (Previously Presented) The apparatus of claim 6, said moving register class fixups comprises one or more of hoisting register class fixups and sinking the register class fixups.

10. (Previously Presented) The apparatus of claim 6, wherein said removing the register class fixups that are unnecessary comprises removing dead code.

11. (**Currently Amended**) A system comprising:

a processor having at least one register; and

a compiler coupled to the processor executing in a host device that inputs a source program having a plurality of operation blocks,

wherein the compiler assigns a first register class in at least one instruction to the at least one symbolic register, and assigns a second register class to the at least one symbolic register through a linear conjunctive forward dataflow analysis that iterates ~~each~~ **a** basic block of instructions **in a compilation unit** only once, moves register class fixups for the assignment of the second register class **to a different location** and removes unnecessary register class fixups to reduce the register class fixups, and renames the at least one symbolic register,

wherein each instruction includes assignment of one of the first register class assigned and the second register class assigned and wherein register class fixups are instructions inserted into a program in response to one or more register class assignment modifications.

12. (Original) The system of claim 11, wherein the first register class assigned is an initially assigned register class.

13. (**Currently Amended**) The system of claim 11, wherein the second register class assigned includes:

marking a register class assignment map that operates to track register class assignments at a block entry of [[a]] **the** compilation unit;

marking [[the]] **a** register class assignment map at a block exit of the compilation unit;

determining [[the]] **a** register class assignment map at an entry of a**n** instruction in a block **of the compilation unit**; and

determining [[the]] **a** register class assignment map at an exit of [[a]] **the** instruction in the block **of the compilation unit**.

14. (Previously Presented) The system of claim 11, said movement of register class fixups includes one or more of:

hoisting register class fixups; and

sinking the register class fixups.

15. (Previously Presented) The system of claim 11, said removing the register class fixups that are unnecessary includes removing dead code.

16. (**Currently Amended**) A computer comprising:

at least one processor having at least one register coupled to a first memory and a second memory;

at least one user input device coupled to the processor;

a monitor coupled to the processor, and

a compiler executing in the processor that inputs a source program having a plurality of operation blocks,

wherein the compiler assigns a first register class in at least one instruction to the at least one symbolic register, and assigns a second register class to the at least one symbolic register through a linear conjunctive forward dataflow analysis that iterates ~~each~~ **a** basic block of instructions **in a compilation unit** only once, moves register class fixups for the assignment of the second register class **to a different location** and removes unnecessary register class fixups to reduce the register class fixups, and renames the at least one symbolic register,

wherein each instruction includes assignment of one of the first register class assigned and the second register class assigned and wherein register class fixups are instructions inserted into a program in response to one or more register class assignment modifications.

17. (Original) The computer of claim 16, wherein the first register class assigned is an initially assigned register class.

18. (**Currently Amended**) The computer of claim 16, wherein the second register class assigned includes:

marking a register class assignment map that operates to track register class assignments at a block entry of [[a]] **the** compilation unit;

marking [[the]] **a** register class assignment map at a block exit of the compilation unit;

determining [[the]] **a** register class assignment map at an entry of a**n** instruction in a block **of the compilation unit**; and

determining [[the]] **a** register class assignment map at an exit of [[a]] **the** instruction in the block **of the compilation unit**.

19. (Previously Presented) The computer of claim 16, said movement of register class fixups includes one or more of:

hoisting register class fixups; and

sinking the register class fixups.

20. (Previously Presented) The computer of claim16, wherein said removing the register class fixups that are unnecessary includes removing dead code.